Homework 2: Arrays

Objective

Introduce the concepts and general practices for programming VerySimpleCPU using our Instruction Set Simulator (ISS).

Background

Bubble sort is a simple sorting algorithm. It means that the sort repeatedly compares pairs of elements and swaps them if needed. Bubble Sort is organized as a series of passes. In each pass, you compare the 1st and 2nd elements and swap if out of order; then the 2nd and 3rd elements; and so on. At the end of the first pass, the smallest element is guaranteed to be at the end of the array.

What To Do

Make sure you have watched this video before you do the lab:
https://youtu.be/5A1j1dzBA0g

Part 1

You have been given a sample assembly code below. The changes in RAM while this code is running are shown. Examine the sample question and answer. Show the RAM changes for the given question. Note: If you have instructions you do not know, you can learn in the InstructionSet_README.txt file. Upload your solution to LMS.

Sample Question: Trace the contents of memory when the below VerySimpleCPU program is run. Express the progression of memory contents in a table.

```
0: CP 50 51          6: LT 101 100        12: 11
1: MUL 50 52         7: BZJ 102 101       50: 5
2: CP 100 50         8: CPi 50 5          51: 4
3: SRLi 100 3        9: ADD 50 51         52: 16
4: ADDi 100 1        10: ADDi 50 1        53: 10
5: CP 101 53         11: BZJi 12 0        102: 10
```

Answer:

PC:		0	1	2	3	4	5	6	7	10	11
50:	5	4	64							65	
100:				64	8	9					
101:							10	0			

Question: Trace the contents of memory when the below VerySimpleCPU program is run. Express the progression of memory contents in a table just like the one above. Create a file called hw2_part1_RAM.txt and write your solution.

```
0: NAND 202 202
1: ADDi 202 1
2: ADD 201 202
3: CP 203 201
4: NANDi 203 9
5: BZJ 200 203
6: NAND 201 201
7: ADDi 201 1
8: BZJi 9 0
9: 8
200: 5
201: 8
202: 3
```

Part 2

1. Make an assembly code multiply by 2 (two) each elements of an array A. The address of A[0] is 500, the address of A[1] is 501. Use CPI and CPIi when accessing A. Call this file hw2_part2.asm.

2. Test it for A[0] = 137 and A[1] = 224. Hence, at the end of your program we will see A[0] = 274 and A[1] = 448.

Part 3

1. Modify bubblesort.asm that sorts the elements of an array in **descending order** to make the assembly work correctly. You will be given comments on the appropriated task to be performed in low-level C, simply translate it into ASM.

Submission

- Submit the following files in LMS under the assignment HW2. Do not zip your files, upload them directly on LMS!
  - hw2_part1_RAM.txt
  - hw2_part2.asm
  - bubblesort.asm